

# Á DEFINIR

*Davi J G Sousa, Alexandre S Vasconcelos*

davi.juvenio@academico.ifpb.edu.br, alexandre.vasconcelos@ifpb.edu.br  
IFPB Campus Campina Grande, Bacharelado em Engenharia de Computação

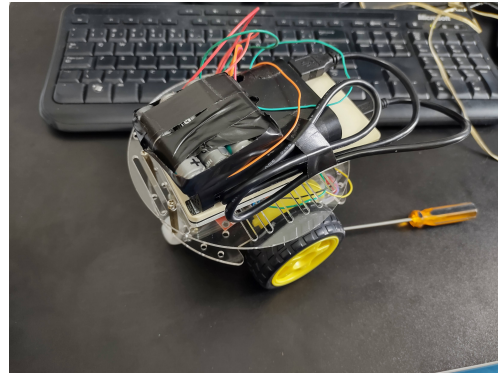
## 1. INTRODUÇÃO

Robôs móveis de escala reduzida, mas com alta capacidade de manobra, são exigidos em terrenos incertos e inalcançáveis como espaço estreito em edifícios danificados após desastres, ambientes de radiação, campos de trabalho complexos em terrenos selvagens, etc [1]. Esses robôs permitem que os pesquisadores obtenham informações relacionadas à distância, e melhorar a eficiência de reconhecimento e capacidade de tratamento de desastres [2]. A primeira missão de resposta quando ocorre um desastre é verificar os danos ao ambiente alvo, incluindo mapeamento no local do desastre [3]. No entanto, o local, incluindo o exterior e o interior dos edifícios, é perigoso para os humanos devido ao constante risco. Portanto, há a necessidade de usar tecnologia robótica móvel para essas missões de exploração, em vez de humanos. Nos últimos anos, robôs móveis equipados com diversos tipos de sensores foram propostos para inspecionar locais de acidentes para obter informações ambientais necessárias e transmiti-las aos socorristas por meio de um sistema de comunicação confiável [4]. A tarefa mais importante é obter as imagens do local do acidente e informações sobre as condições do ambiente para os socorristas. Se não houver perigo, os resgatadores podem avançar e alcançar os robôs e, em seguida, operar o robô para ir ao resgate. Uma vez que as informações ambientais de risco sejam obtidas durante o resgate, ela alertará os socorristas para que façam alguns preparativos. Portanto, isso tornará os socorristas mais seguros. Neste artigo, é apresentado um sistema de robô de busca e resgate DaviBot usado para avaliação remota de ambientes de desastre, que inclui um robô e uma unidade de controle operacional ilustrado na Figura 1.

## 2. OBJETIVOS

Desenvolver um protótipo em escala reduzida de um robô controlado remotamente através de uma comunicação Wi-Fi pelo usuário, através de um computador, que tem por finalidade atuar em áreas ou situações de desastre onde a intervenção humana é perigosa ou impraticável.

**Fig. 1.** DaviBot

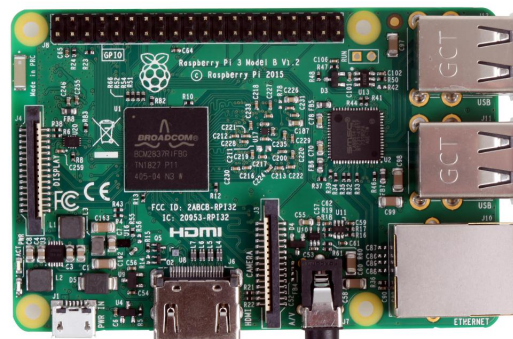


## 3. MATERIAL

### 3.1. Raspberry Pi

A plataforma utilizada no projeto foi a Raspberry Pi 3 Model B, ilustrado na Figura 2, sua escolha foi devido aos requisitos de projeto, pois necessitava de um sistema embarcado que se comunique via Wi-Fi e tenha um poder de processamento considerável, dado os possíveis upgrades do projeto.

**Fig. 2.** Raspberry Pi 3 Model B



### 3.2. Chassis

O modelo reduzido pensado para a realização do projeto foi o ilustrado na Figura 3, que dispõe de uma tração diferen-

cial e duas rodas bobas. Sua escolha vem do fato que essa configuração responde bem com superfícies desregulares.

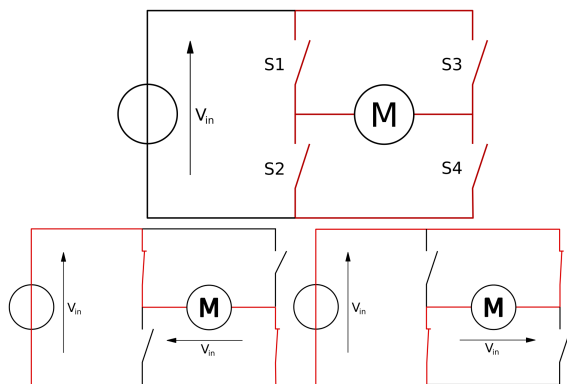
**Fig. 3.** Chassis DaviBot



### 3.3. Ponte H

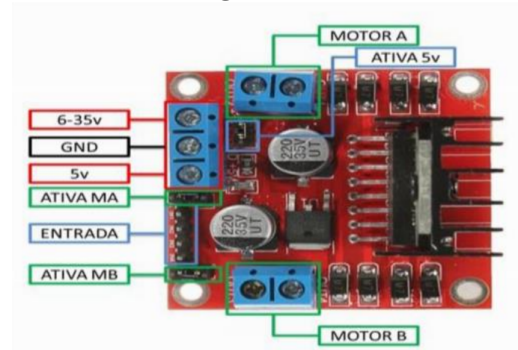
Ponte H ilustrado na Figura 4 é um circuito do tipo chopper de classe E (um chopper de classe E converte uma fonte fixa de corrente continua fixa em uma tensão de corrente continua variável abrindo e fechando diversas vezes), e , portanto, pode determinar o sentido da corrente, a polaridade da tensão e a tensão em um dado sistema ou componente. Acionando-se em conjunto, as chaves S1 e S4, o terminal direito do motor fica com uma tensão mais positiva que o esquerdo, fazendo a corrente fluir da direita para a esquerda. Deste modo, o motor adquire sentido de giro que podemos chamar de Sentido 1. Acionando-se em conjunto as chaves S3 e S2, o terminal esquerdo do motor fica com uma tensão mais positiva que o direito, fazendo a corrente fluir da esquerda para a direita. Deste modo, o motor adquire sentido de giro que podemos chamar de Sentido 2, que é inverso ao Sentido 1.

**Fig. 4.** Ponte H



Para a utilização da ponte H, foi preferível utilizar o modulo ponte H L298N ilustrado na Figura 5, pela facilidade de acesso e disponibilidade no mercado.

**Fig. 5.** L298N



### 3.4. Motor

O motor utilizado foi o motor DC 3-6V, ilustrado na Figura 6, com caixa de redução e eixo duplo. Sua escolha foi devido sua fácil utilização e disponibilidade no mercado. Aqui estão listadas algumas características do motor:

- Tensão de Operação: 3-6VDC
- Torque: 0,35 Kgf/cm (3V) e 0,80 Kgf/cm (6V)
- Sistema magnético com anti-interferência
- Engrenagem com eixo duplo
- Redução: 1:48
- Corrente sem carga:  $\leq 200\text{mA}$  @ 6V,  $\leq 150\text{mA}$  @ 3V
- Velocidade sem carga:  $200 \pm 10\%\text{RPM}$  @ 6V,  $90 \pm 10\%\text{RPM}$  @ 3V
- Dimensões: 70 x 37 x 23mm

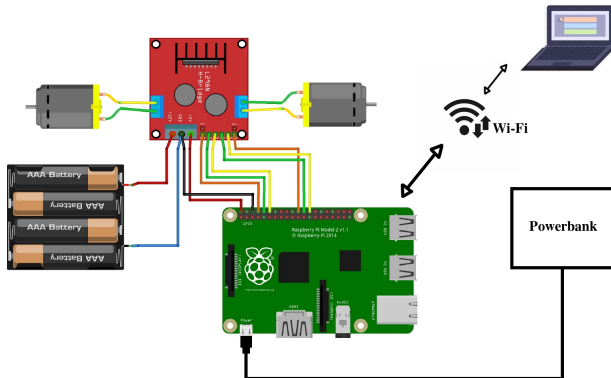
**Fig. 6.** Motor



#### 4. MÉTODO DA PESQUISA

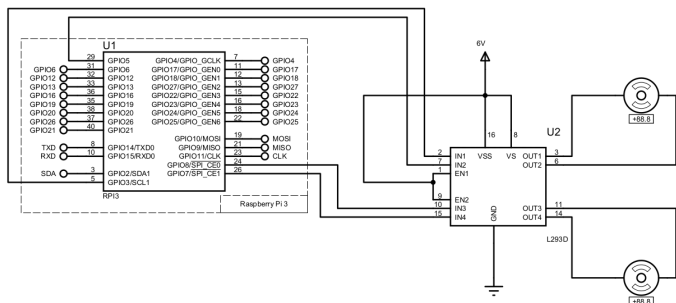
Inicialmente foi realizado a montagem dos componentes, a Figura 7 ilustra o esquema de montagem, onde podemos observar que foi necessário realizar uma comunicação entre a o sistema embarcado Raspberry Pi e um computador via Wi-Fi, através dessa comunicação o usuário acessa a Raspberry Pi remotamente através de um acesso remoto. A partir do sistema embarcado é possível gerar variáveis de controle para movimentar os motores. Essas variáveis de controle é gerada pelos GPIOs da Raspberry Pi, que se comunica com o modulo ponte H, acionando assim suas entradas Enables para movimentação dos motores. As fontes de alimentação para o robô são separadas em duas, a primeira para alimentar a Raspberry Pi por meio de 5V de um powerbank, a segunda por meio de um pack com quatro pilhas de 1,5V, gerando assim uma tenção de 6V.

Fig. 7. Esquemático de montagem



O esquemático do circuito elétrico é apresentado na Figura 8, nele podemos observar que a Raspberry Pi através dos GPIOs 3,5,7 e 8 controla as entradas IN1, IN2, IN3 e IN4 da ponte H L293D para girar os dois motores.

Fig. 8. Esquemático do circuito elétrico



O código do projeto está dividido em 3 partes. parte 1(Ilustrada na Figura 9) onde estão importadas as bibliotecas necessárias para a realização do projeto, os GPIOs de

controle da ponte H(3,5,7 e 8 configurados como saída e inicializados em nível baixo para garantir que o robô não se movimente) e por fim uma mensagem de instrução para o usuário. Parte 2(Ilustrada na Figura 10) onde estão apresentadas as funções frente, trás, direita, esquerda e pare. Parte 3(Ilustrada na Figura 11) onde constantemente espera o comando de entrada do usuário para movimentação do robô.

Fig. 9. Setup

```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(3, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(5, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(7, GPIO.OUT, initial=GPIO.LOW)
GPIO.setup(8, GPIO.OUT, initial=GPIO.LOW)
print("Digite 8 para frente ou 2 para tras ou 4 para esquerda ou 6 para direita ou 5 para parar")
```

Fig. 10. Função

```
def frente():
    GPIO.output(3, GPIO.LOW)
    GPIO.output(5, GPIO.HIGH)
    GPIO.output(7, GPIO.HIGH)
    GPIO.output(8, GPIO.LOW)

def tras():
    GPIO.output(3, GPIO.HIGH)
    GPIO.output(5, GPIO.LOW)
    GPIO.output(7, GPIO.LOW)
    GPIO.output(8, GPIO.HIGH)

def direita():
    GPIO.output(3, GPIO.LOW)
    GPIO.output(5, GPIO.LOW)
    GPIO.output(7, GPIO.HIGH)
    GPIO.output(8, GPIO.LOW)

def esquerda():
    GPIO.output(3, GPIO.LOW)
    GPIO.output(5, GPIO.HIGH)
    GPIO.output(7, GPIO.LOW)
    GPIO.output(8, GPIO.LOW)

def pare():
    GPIO.output(3, GPIO.LOW)
    GPIO.output(5, GPIO.LOW)
    GPIO.output(7, GPIO.LOW)
    GPIO.output(8, GPIO.LOW)
```

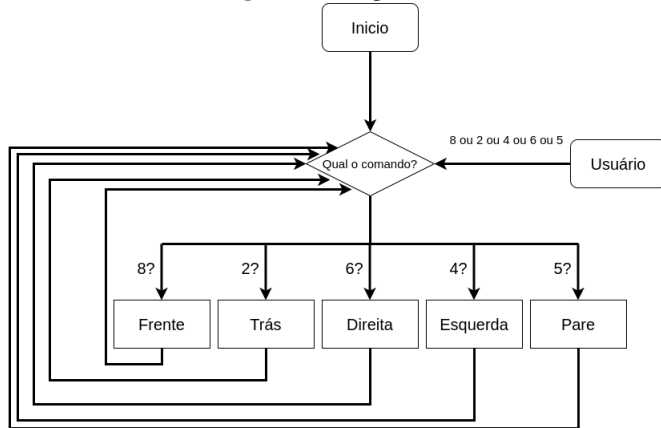
Fig. 11. Repetição

```
while True: # Run forever
    try:
        d=input()
        if d==8:
            frente()
        elif d==2:
            tras()
        elif d==4:
            esquerda()
        elif d==6:
            direita()
        elif d==5:
            pare()
    except:
        print("Voce so poderia ter digitado 8 ou 2 ou 4 ou 6 ou 5")
```

Para melhorar o entendimento, foi gerado um fluxograma do código, ilustrado na Figura 12, onde é possível observar que o programa está constantemente esperando o comando

do usuário, que é um número enviado via terminal do computador do usuário para o sistema embarcado, esse comando é interpretado através de funções que agem no sistema movimentando o robô. A ação é constante e só muda após uma intervenção do usuário, onde escolhe se quer um movimento diferente ou parar o robô.

**Fig. 12.** Fluxograma



O computador do usuário acessa a Raspberry Pi por meio de um acesso remoto, que consiste na capacidade de um computador acessar, à distância, um outro computador ou de uma rede de computadores e, assim, visualizar arquivos e até controlar programas e as funcionalidades dos dispositivos acessados.

## 5. CONCLUSÃO/CONSIDERAÇÕES

O DaviBot cumpriu com os requisitos de projeto, que foi desenvolver um protótipo em escala reduzida de um robô controlado remotamente, esse projeto também tem como objetivo aplicar os conhecimentos da disciplina de sistemas embarcados do Instituto Federal de Campina Grande para obtenção de nota, e cumpriu pois a disciplina seguiu de guia para as constantes barreiras encontradas ao longo do projeto.

As maiores dificuldades no projeto foi a parte prática, pois envolve diversas áreas de conhecimento, como por exemplo mecânica, mecatrônica, engenharia elétrica e engenharia de computação.

Como projetos futuros fica a adição de sensores de localização e processamento de imagem com câmera local, para o usuário poder controlar o robô sem contato visual.

## 6. REFERÊNCIAS

[1] Long Bai, Jian Guan, Xiaohong Chen, Junzhan Hou, and Wenbo Duan, "An optional passive/active transformable wheel-legged mobility concept for search and

rescue robots," *Robotics and Autonomous Systems*, vol. 107, pp. 145–155, 2018.

- [2] Dean F Hougen, Saifallah Benjaafar, Jordan C Bonney, John R Budenske, Mark Dvorak, Maria Gini, Howard French, Donald G Krantz, Perry Y Li, Fred Malver, et al., "A miniature robotic system for reconnaissance and surveillance," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. IEEE, 2000, vol. 1, pp. 501–507.
- [3] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Kazuki Otake, Kazuya Yoshida, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, Mineo Fukushima, et al., "Gamma-ray irradiation test of electric components of rescue mobile robot quince," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE, 2011, pp. 56–60.
- [4] Jingchao Zhao, Junyao Gao, Fangzhou Zhao, and Yi Liu, "A search-and-rescue robot system for remotely sensing the underground coal mine environment," *Sensors*, vol. 17, no. 10, pp. 2426, 2017.